2021-11

Shape Analysis of Flight Trajectories Using Neural Networks

Gingrass, Colton; Singham, Dashi I.; Atkinson, Michael P.

ARC

Colton Gingrass, Dashi I. Singham, and Michael P. Atkinson, "Shape Analysis of Flight Trajectories Using Neural Networks, " JAIS, Vol. 18, No. 11 (2021), pp. 762-773, https://doi.org/10.2514/1.I010923.
https://hdl.handle.net/10945/68100

JOURNAL OF AEROSPACE INFORMATION SYSTEMS
Vol. 18, No. 11, November 2021

Shape Analysis of Flight Trajectories Using Neural Networks

Colton Gingrass,* Dashi I. Singham,† and Michael P. Atkinson‡
Naval Postgraduate School, Monterey, California 93943

https://doi.org/10.2514/1.I010923

The recent widespread implementation of Automatic Dependent Surveillance–Broadcasting (ADS-B) systems on aircraft allows for improved monitoring and air traffic control management. As part of this monitoring, it is important to be able to detect unusual flight trajectories due to weather events, detection avoidance, aircraft malfunction, or other activities that may signal anomalous behavior. Given the large volume of ADS-B data available from aircraft around the world, the ability to automatically determine the shape of the trajectory and identify anomalous behavior is important to reduce the need for human identification and labeling. A neural network model is developed for multicategory classification of the shape of the trajectory using features derived from a large ADS-B data set such as bearing and curvature. The results suggest promise in differentiating common trajectory shapes using key factors, with the accuracy of the classifier being comparable to human accuracy.

I. Introduction

FLIGHT monitoring for anomaly detection is an active area of research. Given the recent widespread deployment of Automatic Dependent Surveillance–Broadcasting (ADS-B) technology, there is a large amount of open-source data available for analyzing aircraft trajectories. This fairly new technology relays information on the aircraft's position, altitude, speed, and other attributes from equipped aircraft in real time with precision. Given that in 2020 the Federal Aviation Administration has standardized and enforced the use of ADS-B on all aircraft in controlled airspace, the large amount of data produced and recorded by this system can be used to establish baseline criteria for flights that deviate from normal air traffic control patterns. Automated methods of classifying and evaluating trajectories can serve to limit the amount of human effort required to study flight behavior, and this has resulted in much research to automatically detect anomalies in flight paths. There are many types of anomalies of interest, for example, aircraft malfunction during takeoff and landing, detours taken to avoid weather events, avoidance of potential collisions for air traffic control, or flights attempting to avoid detection or deviate from their planned trajectory. The ability to use machine learning to automatically flag potentially anomalous flights would allow analysts to narrow their search and quickly diagnose potential problems. The ability to determine intent behind unmanned aerial vehicles using shape analysis is also of interest in civilian and defense communities. For example, certain shapes may signify surveillance or patrolling behavior.

Our focus is on classifying some common and unusual flight trajectory shapes using sparse locational ADS-B data. In our data set, a majority of flights appear to be one-way from the origin to the destination. However, a small percentage of flights may travel in unusual patterns, such as multiple loops, figure-eights, or taking a jagged path toward the destination. Shape classification is important because the shape may be a proxy to detect an underlying activity of interest; for example, a weather event could significantly affect flight trajectories, or the shape may reveal sightseeing or flight training activities. Furthermore, the ability to quickly identify unusual flight trajectories may aid in national security efforts to investigate anomalous behavior across the globe, which often rely on large efforts by human analysts to monitor flight data. Detection avoidance, smuggling, and military maneuvering exercises are all activities of interest that could affect the shape of a flight trajectory.

Our goal is to develop a method for identifying trajectory shape from limited locational data. Because there is a large amount of aviation data available, we wish to be able to quickly classify standard trajectory shapes and identify a small percentage of potentially unusual but distinct shapes. Anomalous paths may fall into a known but infrequent pattern, for example, an aircraft flying in a figure-eight pattern. Alternatively, anomalous trajectories may not fall into any particular category. We define hybrid tracks as those that do not fit into a distinct/definable shape. Aircraft monitoring and intelligence communities are interested in being able to quickly sort through large amounts of flight data to look for unusual behavior signaled by shape, and we develop a method that can be used to find particular shapes with relatively high accuracy. We note that it would be possible to use this approach for trajectory shapes not appearing in our data set if additional data with alternative shapes were available. The general method presented could be applied to a variety of types of shapes and trajectory data, though our specific model applies in the context of the ADS-B data set we studied.

In conducting shape analysis, it is natural to look at the sequence of directional changes made by the moving object. For example, an object traveling in a straight path will have small or zero changes in heading between different observation points. In other trajectories, a sharp turn will register as a single large directional change. While studying the sequence of angles has been used to model shape trajectories in video surveillance or studies of vehicular motion, it appears natural to apply this idea to ADS-B data because these changes in direction can be easily calculated. Studying this "change in heading" isolates the focus on the shape and has the effect of normalizing the observations across different locations, and also normalizing across the general direction a trajectory could take.

We employ a neural network to help account for the underlying complexity in translating from ADS-B information to a shape classification. In addition to including the changes in direction as inputs to the neural network, we use estimated values of the curvature at the observation points in the data. Curvature is a measure of the deviation from a straight line, and can be estimated at each observation using the points immediately before and after the current point. It takes into account change in direction relative to distance traveled, and so provides different information than the change in heading. Both the change in heading and the curvature can be easily estimated at each point using the latitude and longitude coordinates of ADS-B data.

Additionally, we consider many other input features based on the location data, normalized by total track distance, to focus on the shape and not length of the flight. For example, we can include normalized distance between the starting and ending point of the trajectory to distinguish between straight line tracks and loops. We also compare the median location with the halfway location, where the median

Received 21 October 2020; revision received 26 May 2021; accepted for publication 9 August 2021; published online 13 September 2021.

*Ensign, U.S. Navy, Operations Research Department, 1411 Cunningham Road.

†Research Associate Professor, Operations Research Department, 1411 Cunningham Road; dsingham@nps.edu (Corresponding Author).

‡Associate Professor, Operations Research Department, 1411 Cunningham Road.

location is the median latitude and median longitude across all points on the track, and the halfway location is the point on the track where the aircraft has traversed 50% of the track distance. Note that the halfway location must lie on the actual track, whereas the median location need not. For example, a circular track will have a median location in the center of the circle while the halfway point will lie on the circle itself opposite the starting location. Four key locations of interest are the starting point, ending point, halfway point, and the median location. We calculate distributional properties of the normalized distances between these points and all other points on the track, and the distributional properties of the headings between these points and all other points on the track. Such information can be readily calculated and can help isolate different types of shapes.

To the best of our knowledge, our work is the first to apply neural networks to the shape analysis problem in aircraft trajectory using ADS-B data. There are numerous streams of literature to conduct shape analysis of trajectory data as discussed in Sec. II. Many of them involve collecting features of each trajectory, and then using clustering to group together and classify trajectories with similar shapes in an unsupervised setting, whereas our focus is on developing an extensive labeled data set as input to a neural network. Section III describes the data we use and the types of trajectory shapes we want to classify. We classify thousands of tracks falling within nine obvious shape categories (called *standard* tracks) so that the labeled data can be used to train the neural network model. We also consider data that do not fall into one of the standard shapes (*hybrid* tracks), as there can be high variability in flight trajectories. Section III also describes the process for choosing input factors and tuning the hyperparameters of the neural network model. We employ a nearly orthogonal Latin hypercube (NOLH) sampling method for testing various hyperparameters of the neural network to ensure that we cover a space of possible options.

Section IV presents the results of the model. The model performs favorably on the standard data set overall with a weighted F1 score of 88%, and an unweighted average across shape categories of 62%. Given that we observe only 80% accuracy in human labeling due to variability in trajectory appearance, the neural network appears promising in filtering out potentially anomalous behavior. The model is able to clearly differentiate straight-line tracks, which are a significant part of the data set. It becomes harder to differentiate the unusual shapes, but we note that the model performs much better than a random chance assignment, and misclassifications often occur within similar shapes, yielding credence to our choice of inputs. Because much of the ADS-B data fall outside the nine main shapes, we conduct a separate analysis on the hybrid shapes to assess the performance of the trained model on these unclassified trajectories. We find that the model does a relatively good job of separating tracks that generally travel in an outward direction from start to end (one-way trips) from those that return to the starting area (return trips). In some cases, the model is able to further differentiate particularly unusual shapes, and we demonstrate using some real examples. Section V concludes and suggests avenues for future work.

II. Literature Review

We summarize literature related to three main areas. First we discuss papers that use machine learning and neural networks to analyze trajectory data. Next, we focus on research performed on aircraft data for determining anomalous behavior for air traffic control purposes. Third, we describe work performed to analyze the shape of trajectory data, which is of particular interest to us.

There is much work that uses machine learning methods to analyze various types of trajectory prediction problems [1]. Pedestrian movement can be predicted using a Gaussian mixture model [2], video surveillance with anomaly detection [3], velocity and curvature data [4], or autoencoders [5]. We will also use track curvature as one of our input features to detect interesting shapes. Nguyen et al. [6] use a recurrent neural network to train sequence-to-sequence models for predicting where a vessel will arrive based on studying past trajectory data over a space divided into a grid. The model employs automatic identification system (AIS) data and returns the most probable arrival location, in addition to alternative arrival locations. Clustering methods are also a major tool used in motion mapping and prediction (see, e.g., [7]). Trajectory data can be mapped to a feature space, sometimes by using frequency domain methods, and clusters formed to group together similar types of motion.

There has been substantial research on detecting anomalous flight behavior. Many specific techniques have been designed to analyze air data for preventative safety measures in commercial aircraft [8]. Similar to the general motion mapping methods, there has also been a stream of research that uses clustering to group flight trajectories and identifies outliers that may be anomalous, for example, in determining problems during takeoff and landing. Safety events can also be identified using large-scale data sets [9], where both fleet level and flight level anomalies were detected and validated by experts. The method in [10] employs a data transformation stage that consists of taking all information associated with a particular track and converting it to a single vector that can be input to the clustering method. Related work is described in [11–15].

Anomalous behavior in air traffic management is also of high interest. The major method used is clustering to group together similar types of flight patterns and identify anomalous trajectories [16,17]. Flight traffic for air traffic control management for ADS-B data is studied for a specific region in France [18]. The authors first use a clustering algorithm to classify all the traffic flow, and then use an autoencoder to try to determine anomalous behavior. The work in [19] employs random forests to predict whether aircraft rerouting requests will be operationally acceptable by air traffic control officials, whereas the authors of [20] look at new energy metrics using an unsupervised method to identify anomalous behavior in general aviation flights, which have higher safety risks than commercial flights. Unusual arrivals of aircraft to airports can be found using an incremental learning algorithm to perform daily updates to an anomaly detection model [21]. Neural networks have also been used to predict various aspects of flight trajectories, like vertical trajectory [22], traffic flow [23], and automated decisions [24]. More recently, time dependence for air traffic flow can be modeled using recurrent neural networks [25,26].

Finally, we look at research on shape analysis of trajectories. The authors in [27] study video analysis of trajectories by extracting common features from the trajectories and use a clustering algorithm to group together similar types of motion. Anomalous behavior is found by looking for trajectories in a sparse space, or identifying those that behave differently from others in the same cluster. In [28], the authors differentiate between different types of objects on roadways by analyzing features of transformed trajectories, or motion pattern attributes. In air traffic control, Murça et al. [29] employ clustering to classify flights according to arrival patterns while taking into account the complexity of having multiple routes intersecting.

One of the first papers to study the identification of unique trajectory shapes independent of geographical area is [30], which develops a new architecture for modeling and predicting nonlinear motion patterns, in particular circular, polynomial, or sinusoidal patterns. This work relies on the fact that many motion patterns can be modeled using recursive functions. An important stream of work begins in [31], which uses the tangential angles at observation points and models them as a von Mises distribution. The von Mises distribution is a circular distribution and so is a natural choice for modeling directional angles. Building on past work in this area, Guo et al. [32] employ the speed of the object and use kernel density estimation rather than fitting a traditional parametric distribution. They perform unsupervised learning via clustering to detect anomalous trajectories, while employing clustering and information theory.

Along similar lines, Prati et al. [33] and Calderara et al. [34] also model trajectories as a sequence of angles for shape analysis. They employ circular statistics using a mixture of von Mises distributions in order to model the fact that not all angles will have the same properties, and develop an expectation maximization algorithm to estimate the mixture parameters. Then, each sample is assigned to the most similar distribution from the mixture, and the sequences of assignments are run through a clustering algorithm to group and

detect abnormal trajectories. This method was applied to video surveillance data to determine common trajectories of human motion.

Finally, Mcfadyen et al. [35] apply a similar method to aircraft trajectories with the purpose of differentiating between manned and unmanned aircraft, but acknowledge the computational effort required to model the von Mises distribution. Although clusters can be identified, they do not necessarily link to common shapes as observed in general ADS-B data. As we will see, knowledge of additional factors aside from directional angles will greatly improve the classification. Rather than explicitly building a parametric model for the angles via a distribution, the neural network will infer temporal relations between the angles and will employ numerous additional features. As mentioned in Sec. I, the work we propose will be the first to use neural networks for shape prediction in aviation context using ADS-B data. We will employ input features from the literature like heading and curvature, plus add new features based on the starting, ending, and midpoints of the trajectory. We find these additional features help distinguish key shapes in our flight data setting.

III. Neural Network Model Using ADS-B Data

To the best of our knowledge, Gingrass [36] is the first to use supervised learning with neural networks for shape classification using ADS-B data, and this paper formalizes and extends that work. Figure 1 shows the high-level process of our approach. We first analyze the data to determine the types of features that might contribute to classifying the shape, and compute these for each ADS-B track to provide input to the neural network. We also analyze the data to determine the major types of shapes observed, and once the list of shapes has been determined, label the ADS-B tracks to provide output for the neural network to model. The initial steps determined 1000 features, and then a feature reduction step is employed to reduce overfitting and select 200 input features. These features are then used for training the neural network with a NOLH experimental design used to select the model hyperparameters. The final model is then applied to classify the test set and the performance of the model is analyzed. This section describes in detail the data set we used to train the model, the input features collected, and the process used to train the hyperparameters of the neural network model.

A. Data Set

We relied on an ADS-B data set from October 2016 consisting of flights from all over the world. These are civil flights (i.e., nonmilitary) covering various categories, such as passenger, cargo, private, and commercial. There are also a many different types of aircraft in the data set, including various Boeing, Airbus, Embraer, Cessna, Beechcraft, Gulfstream, Piper, and Cirrus models. ADS-B data can be obtained from a variety of sources, including ADS-B Exchange (https://www.adsbexchange.com/) and OpenSky Network (https://opensky-network.org/), and flights can be extracted from raw data using clustering methods such as [37]. We focus on the full flight trajectory from takeoff to landing and filter out flights where we do not have continuous trajectory information for the duration of the flight. Specifically, we remove flights that do not have well-defined takeoff and landing points, which we determine based on location and altitude information at the beginning and end of the trajectory. Furthermore, we remove flights that have time gaps exceeding 20 min during the flight. We also require that flights have at least 10 location observations and travel at least 5 miles. Our final preprocessing step is to reduce the size of the data set: we down-sample the data so that successive observations on a trajectory are at least 2 min apart.

To develop a comprehensive data set to train the network, we manually classified a total of 17,416 flight trajectory shapes. Initially, we went through a total of around 1000 trajectories to form a basis for what the major shapes would be. We identified nine major shapes (called *standard* shapes) and had a tenth label for trajectories that did not fit into the nine standard shapes (called a *hybrid* class). Most trajectories (around 72%) did not fit into an obvious shape and were placed in this hybrid class, and will be studied separately from the initial classifier on the tracks that clearly fit the nine standard shapes (28% of the total). There were 4880 tracks classified as standard, and this data set formed the basis for training the neural network. There could have been many more shapes assigned, but we chose to focus on nine that could be clearly differentiated and appeared enough to be identified. These nine shapes were chosen after going through thousands of trajectories separate from the ones used in neural net calculation multiple times to ensure that all regularly repeating distinct shapes were represented. This leads to a better training process by focusing on a clean data set where the input data meet strict criteria for being included in a standard class.

Examples of these nine standard shapes are displayed in Fig. 2, and we summarize them next. Each figure in this paper displays a modified version of the original track consisting of 20 points equally spaced in distance along the original track. We discuss this 20-point track approximation further in Sec. III.B. Shape 1 is a straight track without any major detours between the start and endpoints. Shape 2 allows for a detour involving one turn, but otherwise is straight. Shape 3 is a curved trajectory, or parabolic in extreme cases. Shape 4 is a single loop, whereas shape 5 consists of multiple loops. Shape 6 is a figure-eight pattern, whereas shape 7 is a flight that goes out to a location and then returns directly back to the origin. Shape 8 is a single switchback pattern that involves two major turns in the middle of the trajectory, and shape 9 is a sinusoidal pattern that includes multiple changes of direction or multiple switchbacks between the start and endpoints. We will call shapes 1–3 *one-way* shapes because they are the most common ways flights traverse from a starting to ending point, whereas shapes 4–9 are *anomalous* shapes, which may signal unusual behavior.

Once we determined these nine shapes of trajectories, we had two of the authors classify a new initial data set (around 600 tracks) to establish a baseline for human accuracy. We found that 21.5% of the trajectories were given different classifications due to varying human interpretations of which category the track belonged to. Given the high variability in the tracks, many appeared to fall somewhere between two of the shapes, and so human interpretation played a role into which shape was assigned, or whether it was deemed *hybrid*. For example, a mostly straight track with slight curvature could be classified as shape 1 (straight) or shape 3 (curved) depending on the

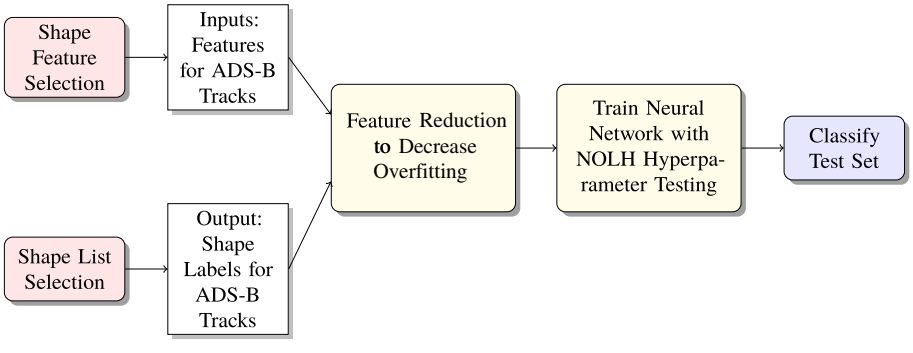
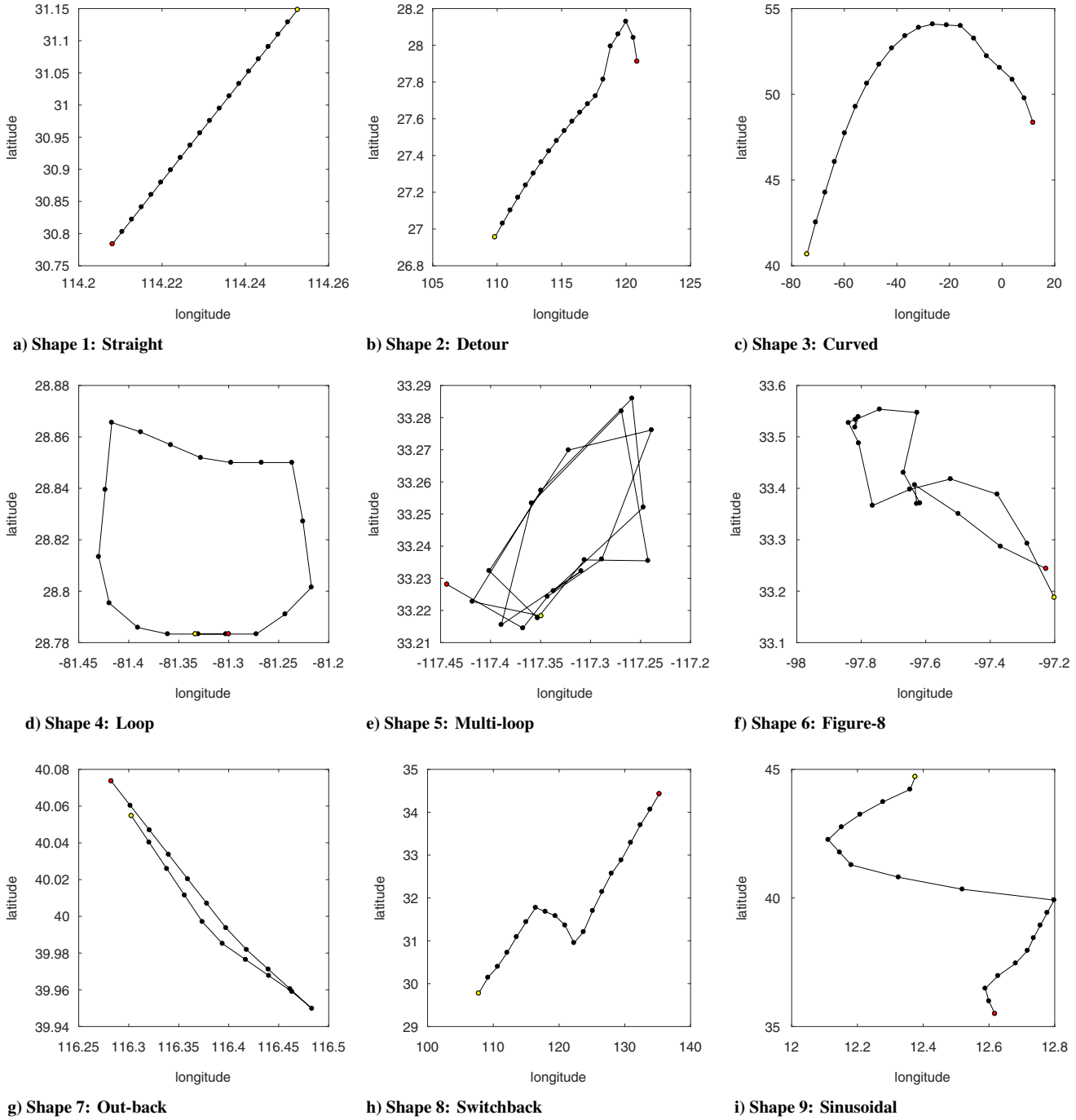


Fig. 1 Process overview.



a) Shape 1: Straight **b) Shape 2: Detour** **c) Shape 3: Curved**

d) Shape 4: Loop **e) Shape 5: Multi-loop** **f) Shape 6: Figure-8**

g) Shape 7: Out-back **h) Shape 8: Switchback** **i) Shape 9: Sinusoidal**

Fig. 2 Major shape categories of tracks.

observer. Similarly, a complex track that returns to its starting point be a combination of shape 4 (loop), shape 6 (figure-eight), or shape 7 (out and back). With additional classification experience and strict guidelines, it would be possible to reduce the human error rate over time. However, due to the variability between observers, in this paper we employed a dataset classified by a single author to reduce the effect of human variability that might lead to conflicting inputs to the model. The assumption is that human labeling will catch nuances that the neural network may not and be more accurate in establishing ground truth. Anything not obviously fitting into one of the nine standard shapes is classified as hybrid. We acknowledge that there may be other sources of human error even from a single expert labeler. However, we achieve continuity by having an author spend a great deal of time studying a large volume of trajectories to define the frequent shapes and label them accordingly after discussing flight patterns with colleagues in aviation.

We further subdivide the hybrid class into those that were somewhere between shapes 1, 2, and 3 (standard one-way trips) and those that were truly anomalous. To avoid training the classifier on ambiguous shapes, we call these hybrid trajectories that were similar to shapes 1–3 as *hybrid-1way* or H1, because they represent common one-way trajectories and 60% of the original 17,416 trajectories fell into this category. Essentially, there were a wide variety of tracks that generally appeared as one-way tracks but could not be obviously labeled as a standard shape due to minor detours, areas of nonstandard curvature, or minor changes of direction at takeoff and landing. The remaining hybrid tracks that were not close to shape 1, 2, or 3 were classified as *hybrid-anomalous*, or HA.

Figure 3 gives examples of trajectories classified as hybrid. The left plot shows one that is hybrid-1way, in that there is both a detour and curvature, which means that it falls between shapes 2 and 3. The right plot shows a trajectory that is hybrid-anomalous, because the

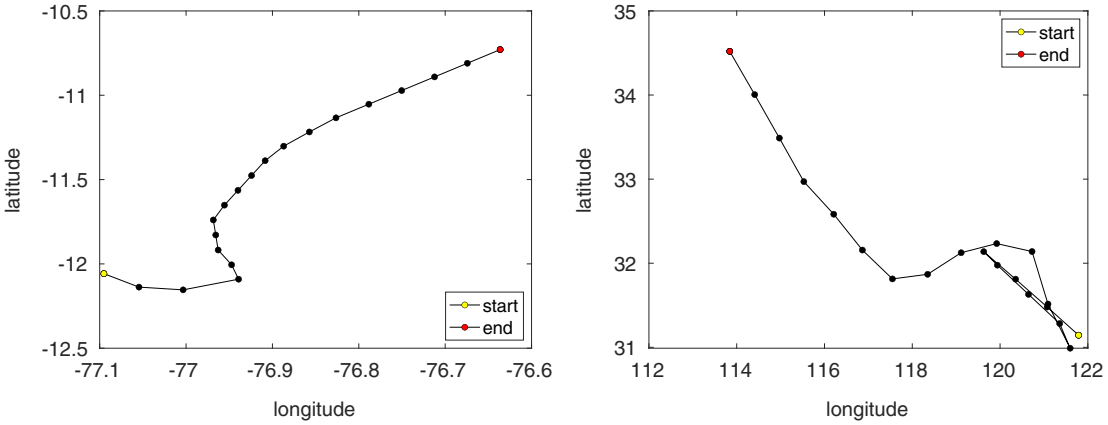


Fig. 3 Examples of hybrid-1way (left) and hybrid-anomalous trajectories (right).

unusual behavior in the bottom right of the figure means that it clearly does not fall into any of the shape categories. The corresponding total track breakdown according to the percentage of observations for the entire data set is in Table 1.

We use the standard data set to train the neural network, and split it into a 70% training set and a 15% validation set for tuning hyperparameters of the network, and display the results of the optimally trained model on a 15% test set that remains unused until the after the model has been finalized.

B. Input Features

Most of our input features relate to heading and curvature, which are key features often used to identify trajectory shape (see Sec. II). The raw data for each ADS-B track are a sequence of latitude and longitude points. From these points we derive 200 input features associated with each track. The first two features are the number of points on the track and total distance traveled. Large values of total distance usually correspond to one-way trips (e.g., shapes 1–3). The remaining features relate to distances, heading, and curvature along the track. We compute distance between points and heading of the aircraft using standard great circle calculations [38]. An aircraft traveling from (lat_1, lon_1) to (lat_2, lon_2) covers a distance in miles of

$$\begin{aligned}\text{dist} = 3958.8 \arccos(&\sin(lat_1)\sin(lat_2) \\ &+ \cos(lat_1)\cos(lat_2)\cos(lon_2 - lon_1))\end{aligned}$$

The initial heading between two points (lat_1, lon_1) and (lat_2, lon_2) is

$$\begin{aligned}\text{heading} = \text{mod}(&\arctan 2(\sin(lon_2 - lon_1)\cos(lat2), \\ &\cos(lat1)\sin(lat2) - (\sin(lat1)\cos(lat2) \\ &\cos(lon_2 - lon_1))) + 360, 360)\end{aligned}$$

Table 1 Classification shapes and percentage of the overall data set

Standard	Hybrid
Shape 1 (straight): 19.72%	Hybrid-1way (H1): 60.42%
Shape 2 (detour): 2.15%	
Shape 3 (curve): 2.88%	
Shape 4 (loop): 0.25%	Hybrid-anomalous (HA): 11.56%
Shape 5 (multiloop): 0.09%	
Shape 6 (figure-eight): 0.44%	
Shape 7 (out and back): 0.26%	
Shape 8 (switchback): 0.91%	
Shape 9 (sinusoidal): 1.32%	

The hybrid-1way (H1) trajectories were similar to shape 1, 2, or 3 but could not be obviously classified as one standard shape.

Our curvature calculation requires three points on a track. We use the Menger variant, which defines curvature as the inverse of the radius of the circle that passes through three points [39]. The three points also define a triangle with side lengths a, b, and c and area $Area$. The circle radius is $abc/4Area$. We normalize this radius by total distance traveled on the track to give the final value of curvature used in our analysis:

$$\text{curvature} = \frac{4Area}{abc} \times \text{track distance}$$

To compute curvature we use Euclidean distance for triangle sides a, b, and c rather than great circle distance because the curvature calculation assumes a Euclidean framework.

Curvature and heading information (specifically how the heading changes) provides valuable information for determining the shape. The heading changes and curvature should be very small for a straight line track (shape 1). A loop track (shape 4) should have roughly constant heading changes and curvature. For an out-back track (shape 7), the heading should remain fairly constant until an abrupt change when the aircraft turns around. The curvature on a figure-eight track (shape 6) should oscillate between large values near the top and bottom of the 8 when the aircraft turns around and smaller values in the middle.

Our first set of features all relate to a modified version of the original track. The modified track consists of 20 points equally spaced in distance along the original track. To create a modified track, we first compute the total distance traveled on the track by summing over the distances of all segments on the track. A segment is just the linearly interpolated portion of the track between two successive observations in the data. We then set the starting (ending) point of the modified track to the starting (ending) point of the original track. The remaining 18 points of the modified track are placed along the original track equidistant apart, using the distance traveled when following the original track. However, the 19 segments created by connecting straight lines between the points are not necessarily of equal length. If the original track has a lot of small turns, then two successive points on the modified track might be close together. This modified track is an approximation with a loss in resolution.

However, having a modified version of our data where every track has the same number of segments provides us with a more consistent avenue to compare tracks. For example, consider two circular-loop tracks: one with 8 equally spaced points and one with 1000. The headings and heading changes will be vastly different between the two tracks even though they are both circles. When using a 20-point approximation for both, the heading calculations will be much closer. We believe that 20 points reasonably capture the major characteristics of a shape. The more points in our approximation, the more input features to our model, which can lead to overfitting. Furthermore, having many small segments might make it difficult to distinguish major directional or curvature changes from minor fluctuations.

Finally the features extracted from our modified tracks constitute only a third of our input features; the remainder are derived from the original tracks.

From the 20-point modified track, we compute the heading between successive points and include the difference between headings at the 18 interior points as features. Additionally, we compute 18 curvature values at these interior points of the track and add them to our feature list. Additional features include summary statistics of the heading differences, absolute heading differences, and curvature. These summary statistics consist of the mean, median, various percentiles, standard deviation, skewness, and kurtosis. Similar to how the von Mises distribution can be used to evaluate the distribution of a sequence of angles, we observe the distribution of our key input variables, which are changes in heading and curvature. Finally, one feature specifies the index (0–19) with the largest absolute heading difference, in an effort to separate shapes with major changes in direction at different points in the trajectory. In total, the modified tracks generate 67 features.

In addition to looking at the change in heading and curvature to determine the shape, we consider numerous other relationships between key locations on the track that might correlate with shape. For the remainder of this section any distance variable is normalized relative to the total distance traveled to focus on the shape of the track. The below features also correspond to the actual track, not the 20-point approximation version described above. Many of the remaining features relate to four locations described in Sec. I:

1) *START*: the starting location of the track.

2) *END*: the ending location of the track.

3) *HALF*: the location on the track where the aircraft has traveled 50% of the total track distance.

4) *MED*: median latitude and longitude across all track points. *MED* is the only location that does not necessarily lie on the track.

Our initial motivation for using these four locations and the features derived from them was distinguishing among three idealized shapes: straight line (shape 1), circular loop (shape 4), and straight out-back (shape 7). We then discovered that these features also did well in classifying other shapes, and thus included them in our final list of input features. As we describe these features below, we highlight how they differ among the shapes.

We define six features corresponding to the distance between all pairs in {*START*, *END*, *HALF*, *MED*}. The distance between *START* and *END* is usually large for one-way tracks (shapes 1, 2, 3, 8, and 9) and small for return tracks (shapes 4, 5, 6, and 7). Even within one-way tracks, the *START* to *END* distance provides differentiation because that distance is close to 1 (recall that we normalize distances relative to total distance traveled) for straight (shape 1) and possibly detours (shape 2), but substantially less than 1 for curved, switchback, and sinusoidal (shapes 3, 8, and 9). The *HALF*-to-*MED* distance will be small for most one-way tracks except for curved (shape 3). For return tracks the *HALF*-to-*MED* distance will be roughly $1/2\pi$ (radius over circumference) for loops (shape 4) and figure-eight (shape 6), $1/2\pi k$ for a k-loop multiloop (shape 5), and 1/4 for out-back (shape 7).

The headings between pairs in {*START*, *END*, *HALF*, *MED*} produce another six features. For straight tracks (shape 1) and detours (shape 2), the heading between *START* and *HALF* will be nearly the same as the heading from *HALF* to *END*. These headings will be substantially different for curved tracks (shape 3) and usually differ by a nontrivial amount for switchback and sinusoidal (shapes 8 and 9). For return tracks (shapes 4–7) the heading between *START* and *HALF* is 180 from the heading from *HALF* to *END*.

For each point on the track, we compute the distance to *START*. We then compute summary statistics for these distances and include them as features. We repeat this for *END*, *HALF*, and *MED* to generate 28 total features. For a straight track (shape 1) the distances corresponding to *MED* follow roughly a uniform distribution on [0,1/2]. For the two loop variants (shapes 4 and 5), the distances corresponding to *MED* (the center of the loop) would be $1/2\pi$ for all points (radius over circumference). While sharing many properties with the loop variants, the figure-eight (shape 6) *MED* distance distribution will not be constant across points. For an out-back track (shape 7), the *MED* distances follow roughly a uniform distribution on [0,1/4].

We generate similar features using heading information rather than distance. For each point on the track, we compute the heading from *START* to that point. We then take the difference between successive headings and calculate summary statistics. We compute similar values for *END*, *HALF*, and *MED* and repeat the process for absolute heading difference to generate 84 more features. For a straight (shape 1) and out-back (shape 7), the heading differences corresponding to *START* would be close to 0 for all points. For detours (shape 2) and switchbacks (shape 8), the *START* heading difference distribution will have a reasonable amount of mass near 0, which corresponds to the part of the track before the detour or switchback. However, detours usually have more mass near 0 as the detour happens near the end of the track, whereas switchbacks often occur near the middle. For the two loop variants (shapes 4 and 5), the heading differences corresponding to *MED* (the center of the loop) would be the same nonnegative constant for all points.

The final seven features relate to curvature across nonsuccessive points. First we compute the curvature between the following three points: the point located at 10% along the total track distance, the point located at 15% along the total track distance, and the point located at 20% along the total track distance. We repeat this for the following six percentile combinations: 45/50/55, 80/85/90, 10/30/50, 50/70/90, 25/50/75, and 10/50/90.

Table 2 summarizes the final feature list by category. The first column specifies whether we derive the features from the original track or the 20-point approximation. The second column lists the metric: distance, curvature, and heading. The third column provides a brief summary of the relevant features, and the final column specifies the number of features in the category.

We originally tried many more input features (around 1000) and were able to obtain similar performance using the 200 presented in Table 2 and so chose the smaller model. In trying to reduce the number of features, we fit a notional neural network on the 1000 features and removed groups of features by category. Using too many input features leads to overfitting, where the training data had very high accuracy in the best fit and the validation set had much lower accuracy. Many of the input features were correlated with each other. For example, we first included the heading between the start, median, and ending point and each point on the 20-point track. However, inclusion of these features let to overfitting the training set, and

Table 2 Summary of input features

Track shape	Metric	Description	No. of features
Original	Number of points on track	— —	1
Original	Total distance traveled along track	— —	1
Original	Distance between {*START*, *END*, *HALF*, *MED*}	Values	6
Original	Heading between {*START*, *END*, *HALF*, *MED* }	Values	6
Original	Distance from {*START*, *END*, *HALF*, *MED* }	Summary statistics	28
Original	Heading differences from {*START*, *END*, *HALF*, *MED*}	Summary statistics	44
Original	Absolute heading differences from {*START*, *END*, *HALF*, *MED*}	Summary statistics	40
Original	Curvature at percentiles on track	Values	7
Approximation	Heading differences	Values; summary statistics	29
Approximation	Absolute heading differences	Summary statistics; max index	11
Approximation	Curvature	Values; summary statistics	27

looking at the heading and curvature information was enough to represent the directionality in the data and lead to improved accuracy in predicting the values in the validation set. Once the feature reduction step was performed, we next moved to fine tuning the hyperparameters of the neural network.

C. Neural Network and Hyperparameter Tuning

Given the complexity of the data set and the potential relationship between the input features and the output shape classifications, we employ a deep sequential neural network to perform supervised learning, specifically a feed-forward neural network. The features described in the previous section are collected for each data observation (trajectory) as inputs to the network. If a feature is collected for each point in the 20-point modified trajectory, then that feature may be represented by 20 inputs per observation for our model. An alternative would be to use recurrent neural networks to model the trajectory using only features from a particular time point, but we can use the simpler feed-forward network by incorporating all time points into the input vector for an observation. The neural network will account for temporal dependence by connecting different features at different times within one input vector.

To tune the network according to relevant hyperparameters, we use a design of experiments method to explore the space of possible networks and choose one with high predictive value by employing a sequence of NOLH experimental designs [40]. NOLH designs are space filling in that they allow for a multidimensional space to be efficiently searched at the expense of pure orthogonality that is obtained in the usual Latin hypercube designs [41]. This means that we do not need to try every combination of possible hyperparameters, but can ensure that we sample from different regions of the multidimensional space.

We need to choose the two major hyperparameters of the network structure: the number of hidden layers (network depth) and the width of each layer (number of neurons). Additionally, we need to choose hyperparameters of the optimization when fitting the network, including the number of epochs to run the optimization, the batch size for the gradient descent, and the learning rate. Finally, to avoid overfitting the network to the training data set, we employ both a dropout rate and an L^2 regularization hyperparameter to obtain a more robust network. We start with broad search ranges over the hyperparameters and assess the performance of the resulting model on the test set.

Based on numerous preliminary NOLH experimental designs, we find four layers to be sufficient to model the complexity of the data, employ a batch size of 256, and choose an L^2 regularization weight of $\lambda = 0.03$. We employ ReLu activation functions for the input layer and three hidden layers, and a softmax activation for the output to predict probabilities that a track falls into one of the nine standard shapes. Fixing these hyperparameters, we run a sequence of refined searches over the remaining four hyperparameters. Figure 4 shows the most refined experimental design used to choose the final network. This particular NOLH design operates by selecting a carefully chosen set of 17 design points that cover the space, while being nearly orthogonal for statistical validity purposes. Thus, we feel comfortable that we have searched a space of possible combinations of hyperparameters to find meaningful interactions between options, rather than performing independent searches for each hyperparameter on an ad hoc basis.

We finalize the remaining hyperparameters by setting the number of neurons in each layer to 475, and the number of epochs to 288. Additionally, we set the dropout rate to 0.9%, and the learning rate to 0.003. This tuning results in 96.6% accuracy in classifying the training set, 88.9% accuracy on the validation set, and 68% unweighted F1 scores across the nine standard shapes on the validation set. Some options tested from Fig. 4 performed much worse, whereas others may have performed comparably in aggregate but at the expense of lower F1 scores in anomalous shapes that do not have many samples. Our goal is to choose a model that will perform well on all standard shapes, rather than simply optimizing for the one-way or shape 1 tracks. In the next section we report specific detailed results applied to the test set (which was not used at all in choosing the hyperparameters). The final network hyperparameters are presented in Table 3.

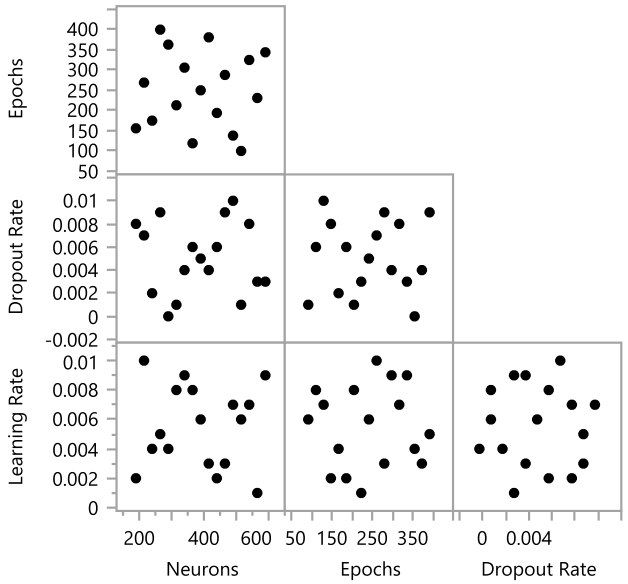


Fig. 4 Refined hyperparameter tuning combinations tested for the number of neurons (width of the network), number of time epochs, the dropout rate, and the learning rate.

Table 3 Hyperparameter values

Hyperparameter	Value
Layers	4
Neurons/layer	475
Learning rate	0.003
Dropout rate	0.009
L^2 regularization	0.03
Batch size	256
Epochs	288

The training process was relatively fast and completed within a few minutes on a personal computer. We built the network using the Keras API, and use Adam optimization with a loss function of categorical cross-entropy, i.e.,

$$Loss = -\sum_{i=1}^{N} w(c_i) \sum_{j=1}^{C} y_{i,j} \log(p_{i,j})$$

where N is the number of total observations in the validation set, C is the number of categories, and c_i is the correct label of observation i. For a given i, let $y_{i,j}$ be zero for all j except with a value of 1 at its correct label when $j = c_i$, and $p_{i,j}$ is the corresponding prediction probability that observation i falls in category j. Additionally, we employ class weights inversely proportional to representation in the training sample size, represented by $w(c_i)$. This discourages overclassifying tracks according to the common shapes by providing higher weight to correct classification of the less frequent shapes.

IV. Classification Results

We first present the results for the trained neural network on standard data in Sec. IV.A. Section IV.B describes how the results of classifier trained on the standard data can be applied to the large volume of hybrid trajectories to identify anomalies.

A. Standard Shape Performance

First, we present the results for standard shapes 1–9 using the trained classifier computed with the model developed in Sec. III.C.

Table 4 shows the precision, recall, and F1 score for each of the shapes applied to the test set that was not used for training or tuning. Some shapes only have a few observations in the test set, so we also perform a fivefold cross-validation on the entire standard data set and present the results in the right-most columns of Table 4.

In general, we will focus the discussion on the cross-validation results because they are not as sensitive to the small sample sizes that occur for some of the shapes in the test set. The classifier has varying performance across shapes, with very strong performance for shape 1 (straight line) with an F1 score of 0.96 for both the test set and cross-validation results. This class also comprises 70% of the data set, so the overall performance of the classifier will be high because of the predominance of shape 1 trajectories. Shape 3 (curved) is the next most common type, and also performs relatively well with an F1 score of 0.83. Shape 2 (single detour), shape 8 (switchback), and shape 9 (sinusoidal) are the next most observed shapes and have comparable F1 scores of 0.63, 0.63, and 0.65, respectively. Presumably, it is easier for the model to train on shapes with more observations. Shape 4 (single loop), shape 5 (multiloop), shape 6 (figure-eight), and shape 7 (out and back) have lower F1 scores, potentially due to infrequent observations, but also for additional reasons considered below related to potential misclassification between shapes.

Table 5 displays the aggregate performance of the classifier applied to the test set, and the cross-validation results. The unweighted averages are the simple averages calculated across the columns in Table 4, whereas the weighted averages are weighted according to the frequency of the shapes. Because of the high frequency of shape 1, the weighted F1 scores are 0.88 for the test set and cross-validation results. However, the unweighted results across shapes reveal an F1 score of 0.62 for the test set, and 0.66 in the cross-validation. This is still significantly better than a random chance classification across the nine standard shapes; however, obtaining more observations to train the less common shapes would potentially improve performance in these shapes.

While Tables 4 and 5 suggest that the classifier will perform significantly better than a random chance draw across nine possible categories, Fig. 5 displays the normalized confusion matrix for the test set and cross-validation results, which yields more insight into the nature of the misclassifications. We see that many of the shape 2 (detour) tracks are misclassified as shape 1, which makes sense because they are mostly straight aside from a single turn. The shape 3 (curved) tracks may be mistaken as straight or detours. The fact that there may be confusion among shapes 1–3 is not surprising because they all travel generally one-way from one location to another. Shape 9 (sinusoidal/jagged) is like shape 2 but with multiple turns, so also has a relatively high misclassification as shape 1. Shape 8 (switchback) is mostly straight, so some will be misclassified as shape 1, but otherwise performs quite well.

One additional source of misclassification between shapes 1, 2, and 3 comes from the way the display projection affected the manual labeling. The authors used the simple equidistant cylindrical map projection, where latitude and longitude are directly converted to Cartesian coordinates to manually classify the tracks. For long international flights this projection may distort the true trajectory shape. For example, a long curved trajectory of shape 3 may correspond to a transatlantic flight that likely flies close to the straight great-circle path. Thus this type of track might be more appropriately classified as shape 1. This distortion will degrade the performance of our algorithm. However, this issue only impacts a relatively small number of trajectories and will primarily affect shapes 1–3 because longer flights are likely one of those shapes.

Table 5 Aggregate performance metrics

	Test set		Fivefold cross-validation	
Metric	Unweighted avg.	Weighted avg.	Unweighted avg.	Weighted avg.
Precision	0.67	0.89	0.68	0.88
Recall	0.61	0.89	0.64	0.88
F1 score	0.62	0.88	0.66	0.88

We also can explain the intuition behind the misclassification of the unusual curved shapes. Shapes 4–7 all have endpoints close to the starting point, so there is some misclassification between these shapes, but not as much between these and shapes 1–3. Shape 5 (multiloop) may be classified as shape 4 (single loop) or shape 6 (a figure-eight) due to high curvature. Shape 4 might be similar to shape 6 or shape 7 (out-back) in that it returns to the starting point once. The difference between shape 6 and shape 7 is mainly that there is an intersection in the path in figure-eights, but not in out-backs, so there will be some misclassifications between the two.

Generally, we find that the classifier can separate out straight tracks (shape 1) with very high precision and recall. It can also differentiate those that travel in a mostly straightforward manner between two start and endpoints from those that return to their starting point. There is very little misclassification between shapes 4–7 and the other shapes. The classifier appears promising in being able to differentiate between different types of curvature, but may require more specific criteria than the input features we used to be able to differentiate the more complicated shapes from each other with high accuracy. Additionally, the tracks reveal high variability relative to the ideal form the shapes would take. The subjective error in human labeling is mainly the result of this variability, so there are natural limits to what we expect the model to predict based on large uncertainty across trajectories that could be classified as the same shape.

B. Hybrid Shape Performance

As most of the tracks were placed in the hybrid class due to the high variation in flight trajectories, we study the effect of applying the trained classifier from the standard data to the hybrid data. We wish to see if the model fit on the standard data can provide inference on the hybrid data that is otherwise difficult to classify. Because much of the hybrid data was *hybrid-1way* (H1) in that it appeared as a standard one-way trip with some variation from shapes 1 to 3, we expect that the classifier trained on standard shapes may still be able to detect anomalous trajectories similar to shapes 4–9. We call those hybrid trajectories that do not appear similar to shapes 1–3 as *hybrid-anomalous* (HA); see Table 1 for a breakdown.

Our objective is to determine how we might tell if a hybrid track is treated differently by the classifier than a standard track. We find that hybrid tracks are less much less likely to be classified as the nominal

Table 4 Category statistics for test set (left) and cross-validation results (right)

	Test set				Fivefold cross-validation			
Class	Precision	Recall	F1	Support	Precision	Recall	F1	Support
Shape 1 (straight)	0.95	0.97	0.96	533	0.95	0.97	0.96	3435
Shape 2 (detour)	0.68	0.59	0.63	58	0.68	0.59	0.63	375
Shape 3 (curved)	0.89	0.71	0.79	66	0.84	0.82	0.83	501
Shape 4 (single loop)	0.50	0.25	0.33	4	0.54	0.49	0.51	43
Shape 5 (multiloop)	1.00	0.50	0.67	2	0.67	0.53	0.59	15
Shape 6 (figure-eight)	0.58	0.58	0.58	12	0.54	0.69	0.61	77
Shape 7 (out-back)	0.20	0.25	0.22	4	0.56	0.43	0.49	46
Shape 8 (switchback)	0.60	0.94	0.73	16	0.75	0.54	0.63	158
Shape 9 (sinusoidal)	0.67	0.70	0.68	37	0.61	0.70	0.65	230

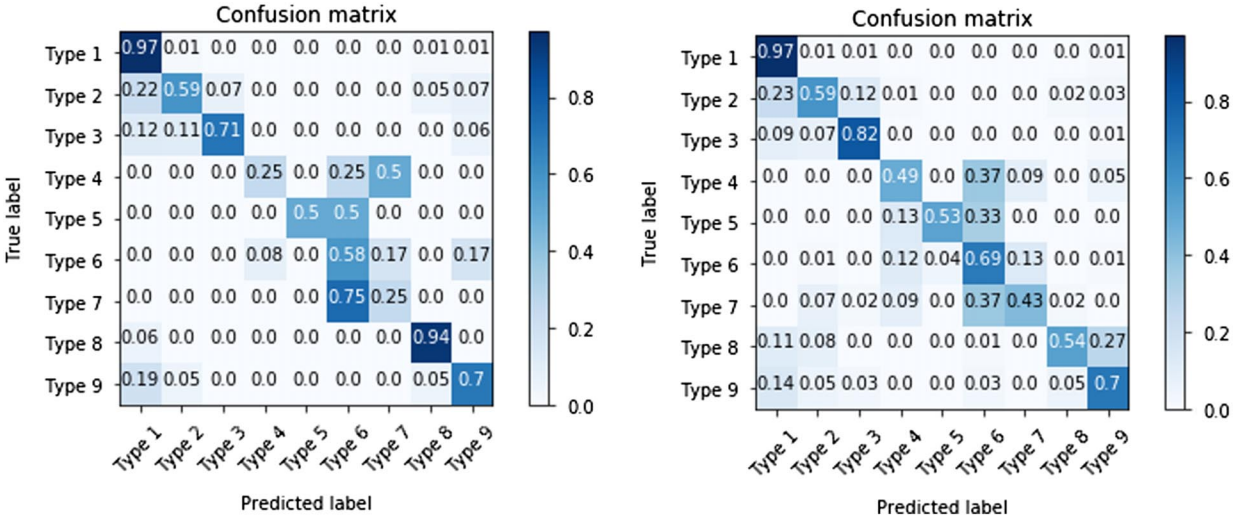


Fig. 5 Normalized confusion matrix for test set (left) and cross-validation (right).

shape 1, and the output probability vector of classification likely has higher entropy than that of the standard tracks. This means that the classifier will have less variability across shapes for standard types that are more likely to fit into a single shape, while distributing the weight more for hybrid tracks that may encompass features of multiple shapes.

Let c_i be the true label of trajectory i. This label can be either an integer 1–9 if the shape is standard, or it can be classified as H1 or HA. The classifier can be applied to a trajectory and will deliver a predicted class $\hat{c}_i$, which can be 1–9. For each track, the prediction is a result of the softmax output layer of the network, where the probability that track i is shape j is computed as $p_{i,j}$ with $\hat{c}_i = \text{argmax}_j p_{i,j}$. The softmax output vector $p_{i,j}, j = 1, \ldots, 9$ contains information on which shapes are most likely for a track, with $\sum_{j=1}^{9} p_{i,j} = 1$. We first look at the overall rate that the classifier assigns tracks to shapes across different data subsets.

Table 6 compares the breakdown of classifications of trajectories across different data sets. First, the labeled training set for standard tracks is broken down across shapes, where we observe that most standard tracks (69.17%) are shape 1. We then show the breakdown of predicted shapes for the validation and test sets. Because these sets should have the same overall distribution of data due to being drawn from the standard data set, they reveal similar breakdowns. A χ^2-goodness-of-fit test reveals a p value of 0.33 when comparing the prediction distributions between the validation and test prediction distributions. However, the validation and test sets have significantly different distributional breakdowns from the training and H1 and HA data sets (χ^2-test p value of 0). This means that the classifier is likely biased; for example, more tracks may be assigned to shape 1 relative to their true proportion (69.17%) in the labeled data, with 74.18 and 72.95% for the validation and test sets, respectively. Unsurprisingly, the breakdowns for H1 and HA will be different from those of the validation and test sets because the properties of the underlying data are fundamentally different.

We next study the columns for H1 and HA in Table 6. In the H1 column, we see that a majority of the tracks are assigned to shapes 1–3 (80.55%), which is reassuring because the one-way trips that did not strictly meet the criteria to be classified as standard will still be labeled as one-way. Of the remaining tracks, 15.69% were classified as shape 9, possibly if there were multiple turns. Only 0.14% were classified as shapes 4–7, which means that even if a track cannot be outright classified as shapes 1–3, it is unlikely to be misclassified as a return trip if is somewhat similar to shapes 1–3. On the other hand, the HA data set has far fewer tracks classified as shapes 1–3 (41.91%), but this still implies that the classifier is likely to have a high false-negative rate in assigning tracks that the human viewed as anomalous as one-way. However, there were a much higher number of tracks assigned to shapes 4–9 than in any of the other data sets. This analysis reveals that the classifier may still be able to identify truly anomalous trajectories even when the track does not fit nicely into one of the standard shapes, at the expense of accidentally classifying anomalous tracks as shapes 1–3.

We can also look at the confidence of the prediction as a measure of quality of the classifier applied to hybrid data. For a given track i, the most focused prediction would put probability 1 in one value of $p_{i,j}$ and 0 for the rest of the shapes j, whereas a random chance guess would put probability 1/9 for each $p_{i,j}$. Table 7 reports metrics on the average strength levels of the prediction for each class. Let N be the total number of data samples in a prediction set, and N_j be the number of trajectories assigned to class j from that particular prediction set where $\mathcal{I}$ is the indicator function so

$$N_j = \sum_{i=1}^{N} \mathcal{I}(\hat{c}_i = j)$$

Table 6 Probability distribution for predictions across shapes

Shape	Training true frequency, %	Predicted frequency, % Validation	Test	H1	HA
Shape 1 (straight)	69.17	74.18	72.95	47.28	13.56
Shape 2 (detour)	7.85	5.05	5.74	17.71	14.75
Shape 3 (curved)	10.77	9.84	9.97	15.56	13.60
Shape 4 (single loop)	0.91	0.41	0.41	0.01	1.29
Shape 5 (multiloop)	0.32	0.14	0.14	0.01	0.70
Shape 6 (figure-eight)	1.64	1.64	1.91	0.11	3.87
Shape 7 (out-back)	1.05	0.96	0.41	0.01	0.45
Shape 8 (switchback)	3.34	3.14	2.60	3.62	12.36
Shape 9 (sinusoidal)	4.95	4.64	5.87	15.69	39.42

Table 7 Average maximum prediction probabilities, and average entropy of predictions (bottom row)

Shape	Standard (test)	H1	HA
Shape 1 (straight)	94.7	84.4	79.4
Shape 2 (detour)	78.1	72.6	76.2
Shape 3 (curved)	86.8	78.1	81.2
Shape 4 (single loop)	85.1	61.0	67.3
Shape 5 (multiloop)	99.6	94.1	92.2
Shape 6 (figure-eight)	75.8	64.7	79.1
Shape 7 (out-back)	75.9	38.8	65.2
Shape 8 (switchback)	78.9	76.2	79.2
Shape 9 (sinusoidal)	78.5	76.0	82.7
Avg. prediction entropy	0.283	0.562	0.539

Then the average maximum prediction for a given class j is

$$\frac{1}{N_j}\sum_{i=1}^{N}\mathcal{I}(\hat{c}_i = j)p_{i,j} \tag{1}$$

where if $\mathcal{I}(\hat{c}_i = j)$ is true, then $p_{i,j} = \max_k p_{i,k}$. Table 7 presents these average maximum prediction values using Eq. (1). For the classifier applied to the standard test set data, we see that the average prediction strengths are quite high, meaning that the classifier is on average giving a preponderance of weight to one class when making a prediction. These values are much lower for H1 than for the test data, meaning that the classifier is not placing these H1 trajectories as strongly in any one class. This makes sense because these tracks do not obviously fit into any of the shapes, and most of them, while appearing one-way, are not clearly in shapes 1–3 according to the human classifier. The HA data set also has lower average prediction strengths for shapes 1–5 and 7. For shapes 6, 8, and 9, the average prediction strength is slightly higher for the HA data, meaning that the classifier may strongly associate these anomalous tracks with some of the anomalous shapes.

Finally, we consider the entropy of the probability predictions for a given track i. These softmax predictions are the values $p_{i,j}$, $j = 1, \ldots, 9$, and the entropy of a distribution is a measure of dispersion. Entropy for a prediction distribution with all its mass at one point is 0, whereas maximum entropy is achieved for the discrete uniform distribution. Entropy for track i can be calculated as

$$-\sum_{j=1}^{9} p_{i,j}\log p_{i,j}$$

The bottom row of Table 7 calculates the average entropy for all prediction probability vectors in the data set. We see that for the standard test data the entropy is 0.283, which is much lower than those for H1 and HA, which are 0.562 and 0.539. This suggests that in addition to the maximum probability assignments being lower on average for hybrid tracks, the probability dispersion across shapes will be higher. This means that the strength of the maximum probability prediction and dispersion levels may be used to identify anomalous tracks that do not fit strongly into any of the shapes. These metrics can separate tracks that strongly match the standard shapes from those that may be in the hybrid class.

We conclude with some illustrative examples of highly anomalous trajectories that are classified by the neural network. In Fig. 6 we consider two trajectories that could be close to figure-eights. The left one is classified as figure-eight with probability 89%, and a second choice of sinusoidal with probability 8.8%, with minimal weights given to loop and out-back. The overall entropy of the classification is 0.418. Understandably, the classifier is affected by the changes in direction and curvature of the track. The plot on the right is classified as a switchback with probability 83.1%, and sinusoidal with 9.6%, with minimal assignments to detour, loop, figure-eight, and out-back. The overall entropy is higher at 0.640. The start and endpoints are far away from each other, and there are multiple sharp changes in direction, which explains why switchback and sinusoidal are chosen as most likely.

Figure 7 has two figures, which are mostly parabolic in shape, with some significant anomalies. The left plot is classified as sinusoidal with probability 96.3%, with smaller probabilities assigned to detour, curved, loop, figure-eight, and switchback. However, the high weight on sinusoidal drives the entropy down to 0.198. The distance between the start and endpoint rules out shapes 4–7, but the extra curvature and

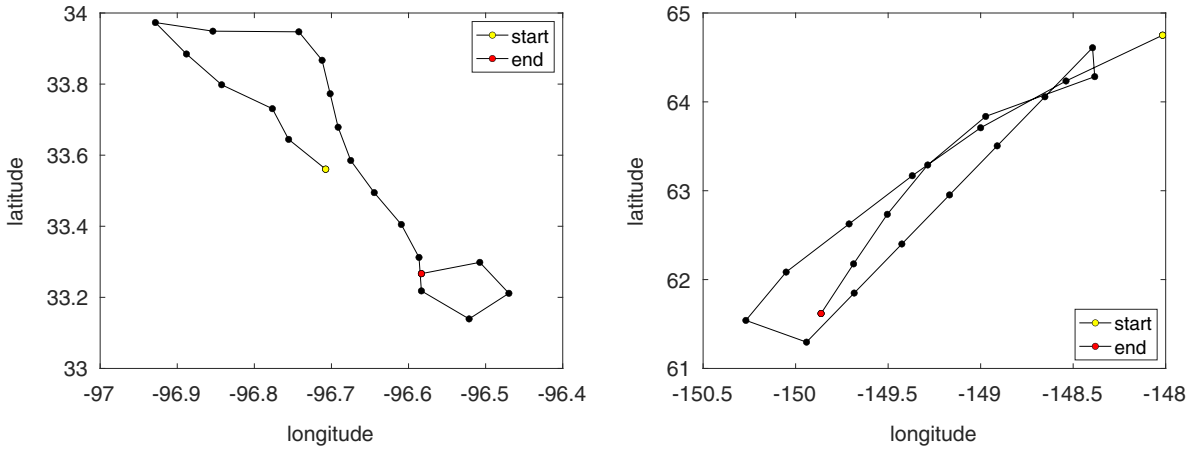


Fig. 6 Left: classified as shape 6 (figure-eight) with probability 89.0% and entropy 0.418. Right: classified as switchback (shape 8) with probability 83.1% and entropy 0.640.

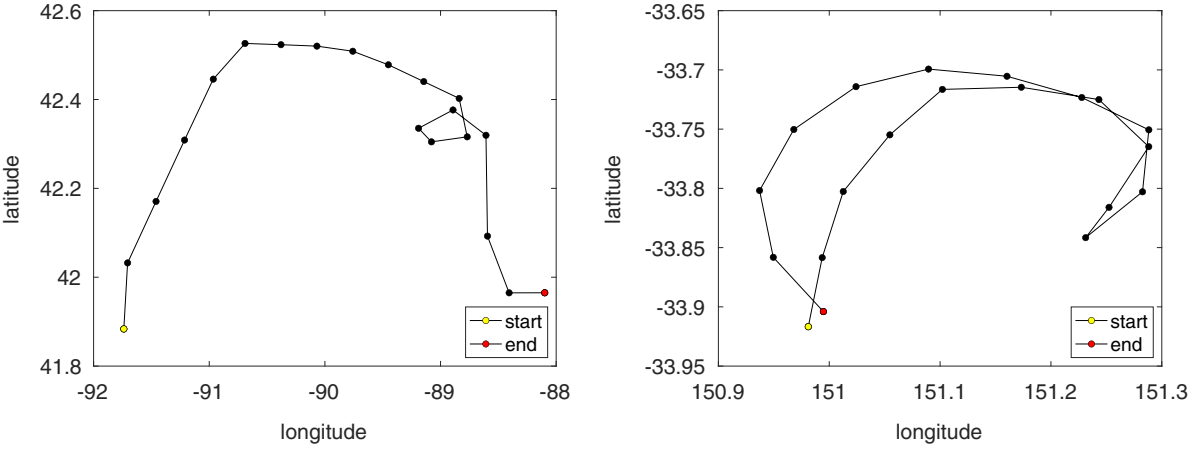


Fig. 7 Left: classified as sinusoidal with probability 96.3% and entropy 0.198. Right: classified as figure-eight with probability 92.4% and entropy 0.341.

changes in direction for the mini-loop correctly prevent the track from being classified as shape 3. The right plot assigns the track as a figure-eight with probability 92.4%, with the remainder of the weight on loop, out-back, and sinusoidal. The overall entropy is 0.341, and the track does display properties of a figure-eight by going out and crossing over itself on the way back. It is possible that the classifier can be used for feature extraction, and although it cannot perform well on highly unusual shapes such as these for which it is untrained, it may be able to spot them as anomalies (i.e., not shapes 1–3).

V. Conclusions

This research represents a first attempt to classify shapes of flight trajectories from ADS-B data using neural networks. Given the large volume of ADS-B data available, it appears possible to train a classifier using human labeled samples to classify a large range of ADS-B trajectories. A neural network is built using as inputs common shape features such as change in heading and curvature, and also consider the distribution between different points in the track that may differentiate shapes. The advantage of using neural networks is that it is possible to quickly train and capture nonlinear relationships using a large number of complex inputs, and thus it is possible to save human effort by identifying potentially anomalous behavior automatically. Because of the high variability present in the shapes of the trajectories (as seen through high error when multiple people label the same data), there are potential benefits of using machine learning to filter out common shapes.

The neural network is trained on a clean dataset of labeled standard trajectories consisting of nine shapes. To tune the network, a design of experiments is employed to explore the space of possible model hyperparameters. The resulting model performs reasonably well on previously unseen test data, with an overall weighted F1 score of 0.88, and an unweighted average across the nine standard shapes of 0.62, implying that the model is reasonably robust across different trajectory shapes. Cross-validation efforts over the entire standard data set deliver similar results. The classifier is also applied to trajectories that do not fit cleanly into the nine standard shapes (hybrid shapes), and this demonstrates that the classifier may still be able to isolate anomalous trajectories from those that are mostly straight or one-way for these types of unclassified data. The model will likely provide stronger predictions for trajectories more closely meeting the standard shapes, either through high maximum probability assignments or lower entropy.

The classifier does a good job separating one-way from return trips, and shapes with minimal curvature from those with large amounts of curvature. Future efforts could spend more time closely defining anomalous shapes of interest and collecting or simulating more types of these tracks to provide additional data for the model to train on. Furthermore, the location or altitude of the track was not considered, and these are features that may correlate with the shape. A small sample of location values (only 20 points per track) is used for the change in heading and curvature values, which allowed for each trajectory to be represented by the same number of input features. However, using more of the information available in each track may yield potentially useful inputs, or allow von Mises distributions to be fit on sequences of angles. Given the robustness of the model presented on the inputs chosen and the speed of training the network, there are numerous other inputs that could be used to identify specific anomalies, and neural networks continue to be a promising tool for quickly automating the classification process.

The general approach of using feed-forward neural networks may be generally applicable to other types of shape data. However, the set of shapes along with the specific input features used would be probably need to be different. For example, pedestrian trajectories may exhibit corners rather than curves, and curvature data might not be as helpful in classifying walking trajectories. Drone data might feature hovering or loitering in place, and hence may require different shapes to represent these activities of interest. The success of the method on a different type of data would depend on careful visual analysis of many trajectories in order to determine the best possible predictors.

Acknowledgement

We are extremely grateful to the Center for Multi-Intelligence Studies at the Naval Postgraduate School for their support of this work.

References

[1] Georgiou, H., Karagiorgou, S., Kontoulis, Y., Pelekis, N., Petrou, P., Scarlatti, D., and Theodoridis, Y., "Moving Objects Analytics: Survey on Future Location & Trajectory Prediction Methods," arXiv preprint arXiv:1807.04639, 2018, https://arxiv.org/abs/1807.04639.

[2] Sillito, R. R., and Fisher, R. B., "Semi-Supervised Learning for Anomalous Trajectory Detection," *Proceedings of the 19th British Machine Vision Conference*, Vol. 1, 2008, pp. 1035–1044, http://www.bmva.org/bmvc/2008/papers/289.html.

[3] Morris, B. T., and Trivedi, M. M., "A Survey of Vision-Based Trajectory Learning and Analysis for Surveillance," *IEEE Transactions on Circuits and Systems for Video Technology*, Vol. 18, No. 8, 2008, pp. 1114–1127. https://doi.org/10.1109/TCSVT.2008.927109

[4] Junejo, I. N., Javed, O., and Shah, M., "Multi Feature Path Modeling for Video Surveillance," *Proceedings of the 17th International Conference on Pattern Recognition, 2004. ICPR 2004*, Vol. 2, Inst. of Electrical and Electronics Engineers, New York, 2004, pp. 716–719. https://doi.org/10.1109/ICPR.2004.1334359

[5] Boyle, J., Nawaz, T., and Ferryman, J., "Deep Trajectory Representation-Based Clustering for Motion Pattern Extraction in Videos," *2017 14th IEEE International Conference on Advanced Video and Signal Based Surveillance (AVSS)*, Inst. of Electrical and Electronics Engineers, New York, 2017, pp. 1–6. https://doi.org/10.1109/AVSS.2017.8078509

[6] Nguyen, D.-D., Le Van, C., and Ali, M. I., "Vessel Trajectory Prediction Using Sequence-to-Sequence Models over Spatial Grid," *Proceedings of the 12th ACM International Conference on Distributed and Event-Based Systems*, Assoc. for Computing Machinery, New York, 2018, pp. 258–261. https://doi.org/10.1145/3210284.3219775

[7] Nawaz, T., Cavallaro, A., and Rinner, B., "Trajectory Clustering for Motion Pattern Extraction in Aerial Videos," *2014 IEEE International Conference on Image Processing (ICIP)*, Inst. of Electrical and Electronics Engineers, New York, 2014, pp. 1016–1020. https://doi.org/10.1109/ICIP.2014.7025203

[8] Smart, E., Brown, D., and Denman, J., "A Two-Phase Method of Detecting Abnormalities in Aircraft Flight Data and Ranking Their Impact on Individual Flights," *IEEE Transactions on Intelligent Transportation Systems*, Vol. 13, No. 3, 2012, pp. 1253–1265. https://doi.org/10.1145/3210284.3219775

[9] Matthews, B., Das, S., Bhaduri, K., Das, K., Martin, R., and Oza, N., "Discovering Anomalous Aviation Safety Events Using Scalable Data Mining Algorithms," *Journal of Aerospace Information Systems*, Vol. 10, No. 10, 2013, pp. 467–475. https://doi.org/10.2514/1.I010080

[10] Li, L., Das, S., John Hansman, R., Palacios, R., and Srivastava, A. N., "Analysis of Flight Data Using Clustering Techniques for Detecting Abnormal Operations," *Journal of Aerospace Information Systems*, Vol. 12, No. 9, 2015, pp. 587–598. https://doi.org/10.2514/1.I010329

[11] Chidester, T. R., "Understanding Normal and Atypical Operations Through Analysis of Flight Data," *Proceedings of the 12th International Symposium on Aviation Psychology*, 2003, pp. 239–242, https://trid.trb.org/view/1104478.

[12] Li, L., Gariel, M., Hansman, R. J., and Palacios, R., "Anomaly Detection in Onboard-Recorded Flight Data Using Cluster Analysis," *2011 IEEE/AIAA 30th Digital Avionics Systems Conference*, Inst. of Electrical and Electronics Engineers, New York, 2011, pp. 4A4-1–4A4-11. https://doi.org/10.1109/DASC.2011.6096068

[13] Li, L., "Anomaly Detection in Airline Routine Operations Using Flight Data Recorder Data," Ph.D. Thesis, Massachusetts Inst. of Technology, Cambridge, MA, 2013.

[14] Li, L., Hansman, R. J., Palacios, R., and Welsch, R., "Anomaly Detection via a Gaussian Mixture Model for Flight Operation and Safety Monitoring," *Transportation Research Part C: Emerging Technologies*, Vol. 64, March 2016, pp. 45–57. https://doi.org/10.1016/j.trc.2016.01.007

[15] Li, G., Lee, H., Rai, A., and Chattopadhyay, A., "Analysis of Operational and Mechanical Anomalies in Scheduled Commercial Flights Using a Logarithmic Multivariate Gaussian Model," *Transportation Research Part C: Emerging Technologies*, Vol. 110, Jan. 2020,

pp. 20–39.
https://doi.org/10.1016/j.trc.2019.11.011
[16] Song, Y., Cheng, P., and Mu, C., "An Improved Trajectory Prediction Algorithm Based on Trajectory Data Mining for Air Traffic Management," *2012 IEEE International Conference on Information and Automation*, Inst. of Electrical and Electronics Engineers, New York, 2012, pp. 981–986.
https://doi.org/10.1109/ICInfA.2012.6246959
[17] Gariel, M., Srivastava, A. N., and Feron, E., "Trajectory Clustering and an Application to Airspace Monitoring," *IEEE Transactions on Intelligent Transportation Systems*, Vol. 12, No. 4, 2011, pp. 1511–1524.
https://doi.org/10.1109/TITS.2011.2160628
[18] Olive, X., and Basora, L., "Identifying Anomalies in Past En-Route Trajectories with Clustering and Anomaly Detection Methods," *Thirteenth USA/Europe Air Traffic Management Research and Development Seminar*, Paper hal-02345597, 2019, http://www.atmseminar.org.
[19] Evans, A. D., Lee, P., and Sridhar, B., "Predicting the Operational Acceptance of Airborne Flight Reroute Requests Using Data Mining," *Transportation Research Part C: Emerging Technologies*, Vol. 96, Nov. 2018, pp. 270–289.
https://doi.org/10.1016/j.trc.2018.09.024
[20] Puranik, T. G., and Mavris, D. N., "Anomaly Detection in General-Aviation Operations Using Energy Metrics and Flight-Data Records," *Journal of Aerospace Information Systems*, Vol. 15, No. 1, 2018, pp. 22–36.
https://doi.org/10.2514/1.I010582
[21] Deshmukh, R., and Hwang, I., "Incremental-Learning-Based Unsupervised Anomaly Detection Algorithm for Terminal Airspace Operations," *Journal of Aerospace Information Systems*, Vol. 16, No. 9, 2019, pp. 362–384.
https://doi.org/10.2514/1.I010711
[22] Le Fablec, Y., and Alliot, J.-M., "Using Neural Networks to Predict Aircraft Trajectories," *Proceedings of the International Conference on Artificial Intelligence, IC-AI'99*, CSREA Press, 1999, pp. 524–529, https://dblp.org/rec/conf/icai/1999-2.html.
[23] Cheng, T., Cui, D., and Cheng, P., "Data Mining for Air Traffic Flow Forecasting: A Hybrid Model of Neural Network and Statistical Analysis," *Proceedings of the 2003 IEEE International Conference on Intelligent Transportation Systems*, Vol. 1, Inst. of Electrical and Electronics Engineers, New York, 2003, pp. 211–215.
https://doi.org/10.1109/ITSC.2003.1251950
[24] Kulkarni, V. B., "Intelligent Air Traffic Controller Simulation Using Artificial Neural Networks," *2015 International Conference on Industrial Instrumentation and Control (ICIC)*, Inst. of Electrical and Electronics Engineers, New York, 2015, pp. 1027–1031.
https://doi.org/10.1109/IIC.2015.7150897
[25] Liu, H., Lin, Y., Chen, Z., Guo, D., Zhang, J., and Jing, H., "Research on the Air Traffic Flow Prediction Using a Deep Learning Approach," *IEEE Access*, Vol. 7, Inst. of Electrical and Electronics Engineers, New York, 2019, pp. 148,019–148,030.
https://doi.org/10.1109/ACCESS.2019.2945821
[26] Shi, Z., Xu, M., Pan, Q., Yan, B., and Zhang, H., "LSTM-Based Flight Trajectory Prediction," *2018 International Joint Conference on Neural Networks (IJCNN)*, Inst. of Electrical and Electronics Engineers, New York, 2018, pp. 1–8.
https://doi.org/10.1109/IJCNN.2018.8489734
[27] Anjum, N., and Cavallaro, A., "Multifeature Object Trajectory Clustering for Video Analysis," *IEEE Transactions on Circuits and Systems for Video Technology*, Vol. 18, No. 11, 2008, pp. 1555–1564.
https://doi.org/10.1109/TCSVT.2008.2005603
[28] Zaki, M. H., and Sayed, T., "A Framework for Automated Road-Users Classification Using Movement Trajectories," *Transportation Research Part C: Emerging Technologies*, Vol. 33, Aug. 2013, pp. 50–73.
https://doi.org/10.1016/j.trc.2013.04.007
[29] Murça, M. C. R., Hansman, R. J., Li, L., and Ren, P., "Flight Trajectory Data Analytics for Characterization of Air Traffic Flows: A Comparative Analysis of Terminal Area Operations Between New York, Hong Kong and Sao Paulo," *Transportation Research Part C: Emerging Technologies*, Vol. 97, Dec. 2018, pp. 324–347.
https://doi.org/10.1016/j.trc.2018.10.021
[30] Tao, Y., Faloutsos, C., Papadias, D., and Liu, B., "Prediction and Indexing of Moving Objects with Unknown Motion Patterns," *Proceedings of the 2004 ACM SIGMOD International Conference on Management of Data*, Assoc. for Computing Machinery, New York, 2004, pp. 611–622.
https://doi.org/10.1145/1007568.1007637
[31] Guo, Y., Xu, Q., Yang, Y., Liang, S., Liu, Y., and Sbert, M., "Anomaly Detection Based on Trajectory Analysis Using Kernel Density Estimation and Information Bottleneck Techniques," Univ. of Girona TR 108, Girona, Spain, 2014.
[32] Guo, Y., Xu, Q., Li, P., Sbert, M., and Yang, Y., "Trajectory Shape Analysis and Anomaly Detection Utilizing Information Theory Tools," *Entropy*, Vol. 19, No. 7, 2017, Paper 323.
https://doi.org/10.3390/e19070323
[33] Prati, A., Calderara, S., and Cucchiara, R., "Using Circular Statistics for Trajectory Shape Analysis," 2008 *IEEE Conference on Computer Vision and Pattern Recognition*, Inst. of Electrical and Electronics Engineers, New York, 2008, pp. 1–8.
https://doi.org/10.1109/CVPR.2008.4587837
[34] Calderara, S., Prati, A., and Cucchiara, R., "Mixtures of von Mises Distributions for People Trajectory Shape Analysis," *IEEE Transactions on Circuits and Systems for Video Technology*, Vol. 21, No. 4, 2011, pp. 457–471.
https://doi.org/10.1109/TCSVT.2011.2125550
[35] Mcfadyen, A., O'Flynn, M., Martin, T., and Campbell, D., "Aircraft Trajectory Clustering Techniques Using Circular Statistics," *2016 IEEE Aerospace Conference*, Inst. of Electrical and Electronics Engineers, New York, 2016, pp. 1–10.
https://doi.org/10.1109/AERO.2016.7500601
[36] Gingrass, C., "Classifying ADS-B Trajectory Shapes Using a Dense Feed-Forward Neural Network," Master's Thesis, Naval Postgraduate School, Monterey, CA, 2020.
[37] Sun, J., Ellerbroek, J., and Hoekstra, J., "Flight Extraction and Phase Identification for Large Automatic Dependent Surveillance–Broadcast Datasets," *Journal of Aerospace Information Systems*, Vol. 14, No. 10, 2017, pp. 566–572.
https://doi.org/10.2514/1.I010520
[38] Williams, E., "Aviation Formulary V1.46," 2011, https://www.edwilliams.org/avform.htm.
[39] Léger, J.-C., "Menger Curvature and Rectifiability," *Annals of Mathematics*, Vol. 149, No. 3, 1999, pp. 831–869.
https://doi.org/10.2307/121074
[40] Sanchez, S., "NOLHdesigns Spreadsheet," 2011, http://harvest.nps.edu/10.2307/121074.
[41] Cioppa, T. M., and Lucas, T. W., "Efficient Nearly Orthogonal and Space-Filling Latin Hypercubes," *Technometrics*, Vol. 49, No. 1, 2007, pp. 45–55.
https://doi.org/10.1198/004017006000000453

D. Casbeer
Associate Editor